I dag

Visualisering

- Diagram
- 3D-grafer
- Matriser och bilder (Orientering. Läs själv.)

Histogram och stapeldiagram

- Med kommandona `bar`, `stem` och `hist` kan man rita *stapeldiagram*, *stolpdiagram* respektive *histogram*.
- *Syntax.*

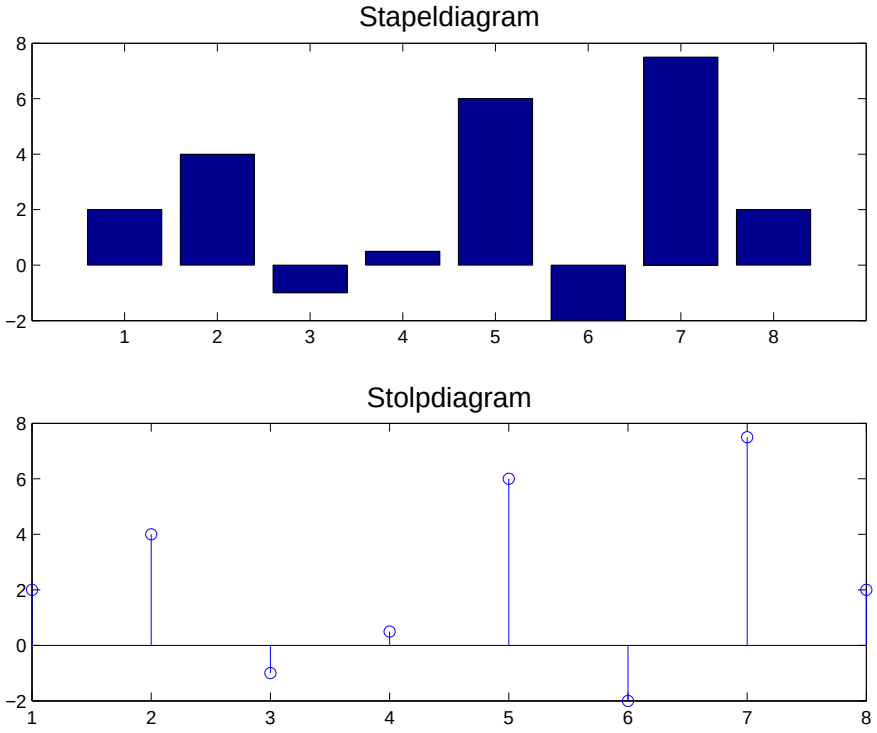
`bar(x)`	Ritar ett stapeldiagram över elementen i x.
`stem(x)`	Ritar ett stolpdiagram över elementen i x.
`hist(x)`	Ritar ett histogram med 10 klasser över elementen i x.
`hist(x,n)`	Ritar ett histogram med n klasser över elementen i x.
`[m,k] = hist(x)`	Ger en vektor k med de 10 klassernas mittpunkter, och en vektor m med antalet element i varje klass.
`[m,k] = hist(x,n)`	Ger en vektor k med de n klassernas mittpunkter, och en vektor m med antalet element i varje klass.

Histogram och stapeldiagram (forts.)

- *Exempel.*

```
>> x = [2 4 -1 0.5 6 -2 7.5 2];

>> subplot(2,1,1)
>> bar(x)
>> title('Stapeldiagram','FontSize',15)

>> subplot(2,1,2)
>> stem(x)
>> title('Stolpdiagram','FontSize',15)
```

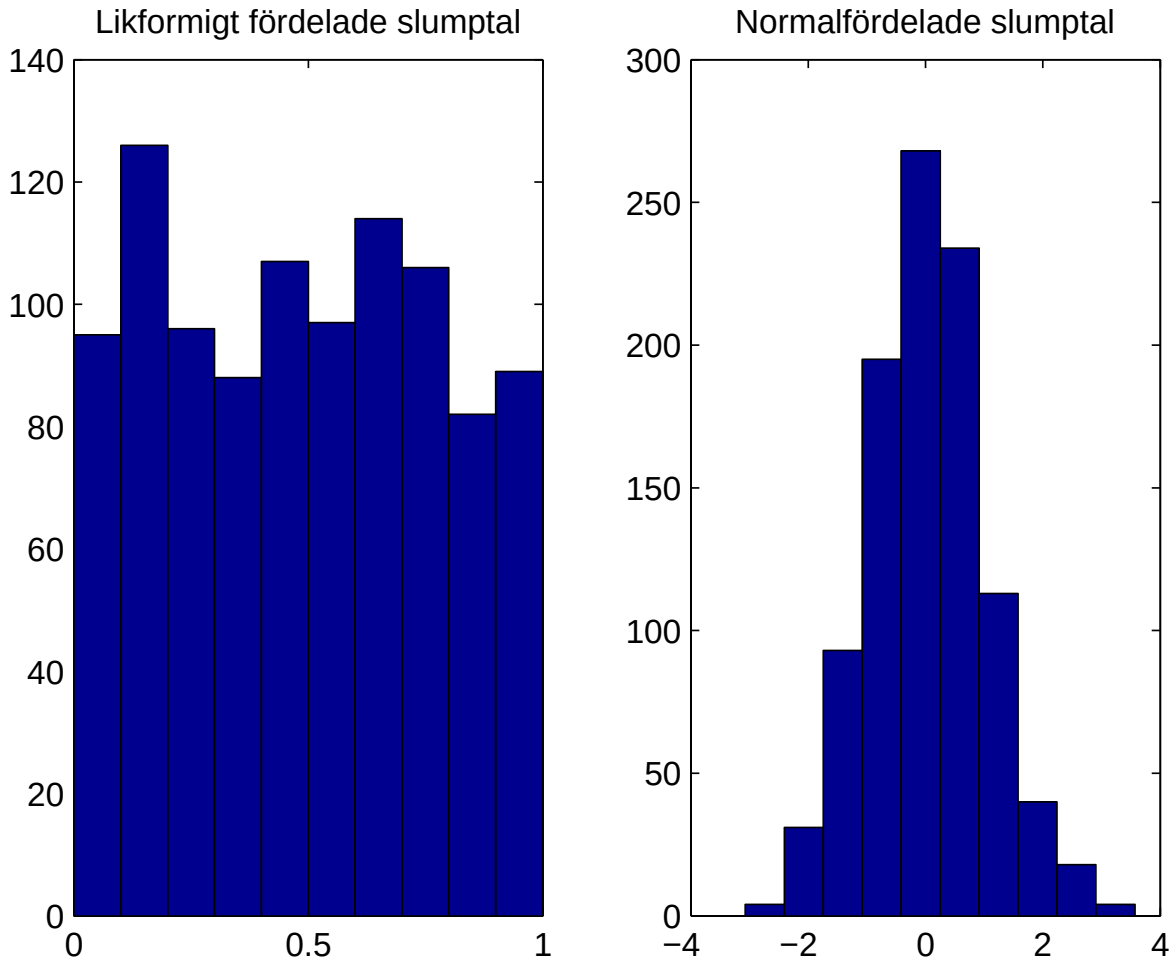
Histogram och stapeldiagram (forts.)



Histogram och stapeldiagram (forts.)

- *Exempel.*

```
>> x1 = rand(1,1000); % Ger en 1 x 1000-matris med likformigt
                      % fördelade slumptal mellan 0 och 1.
>> x2 = randn(1,1000); % Funktionen randn ger normalfördelade
      % slumptal med medelvärde = 0 och standardavvikelse = 1.

>> [m1,k1] = hist(x1,10); [m2,k2] = hist(x2,10);
>> m1, m2    % Antal element i varje klass.
m1 =
    95   126    96    88   107    97   114   106    82    89
m2 =
     4    31    93   195   268   234   113    40    18     4

>> subplot(1,2,1), hist(x1,10), title('Likformigt fördelade slumptal')
>> subplot(1,2,2), hist(x2,10), title('Normalfördelade slumptal')
```


Likformigt fördelade slumptal
140
120
100
80
60
40
20
0
0
0.5
1
Normalfördelade slumptal
300
250
200
150
100
50
0
−4
−2
0
2
4

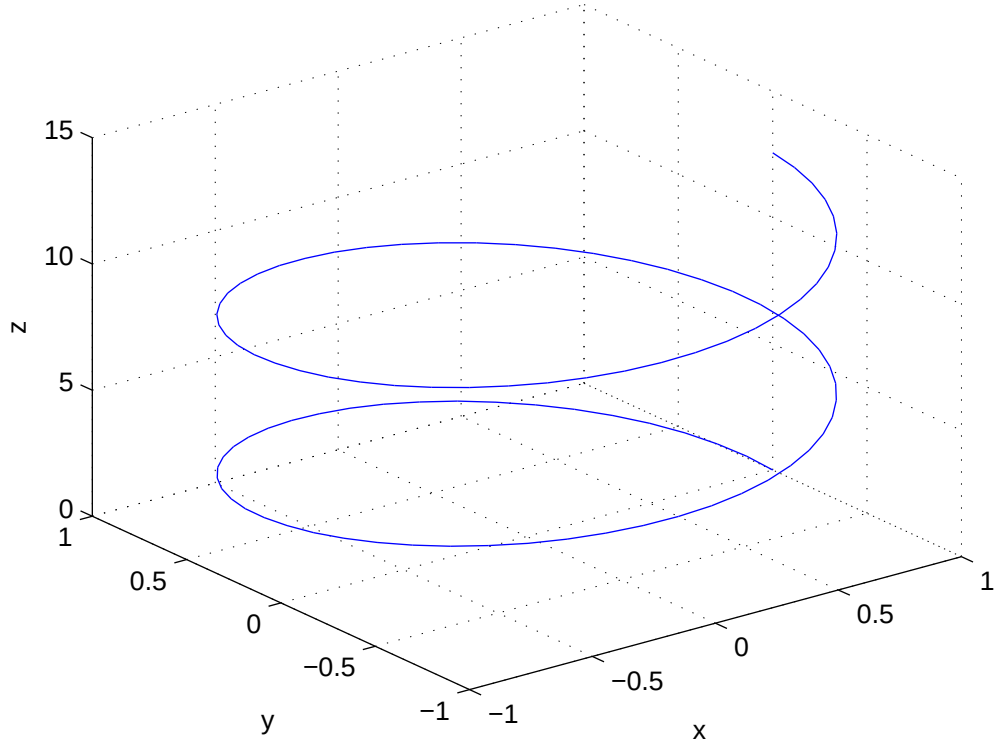
3D-grafer

- *Exempel.* Plotta kurvan (spiralfjädern):

$$\begin{cases} x = \cos(t) \\ y = \sin(t) \\ z = t \end{cases}, \quad 0 \leq t \leq 4\pi.$$

```
>> t = linspace(0,4*pi);

>> x = cos(t); y = sin(t); z = t;

>> plot3(x,y,z)
>> xlabel('x'), ylabel('y'), zlabel('z'), grid
```

3D-grafer (forts.)



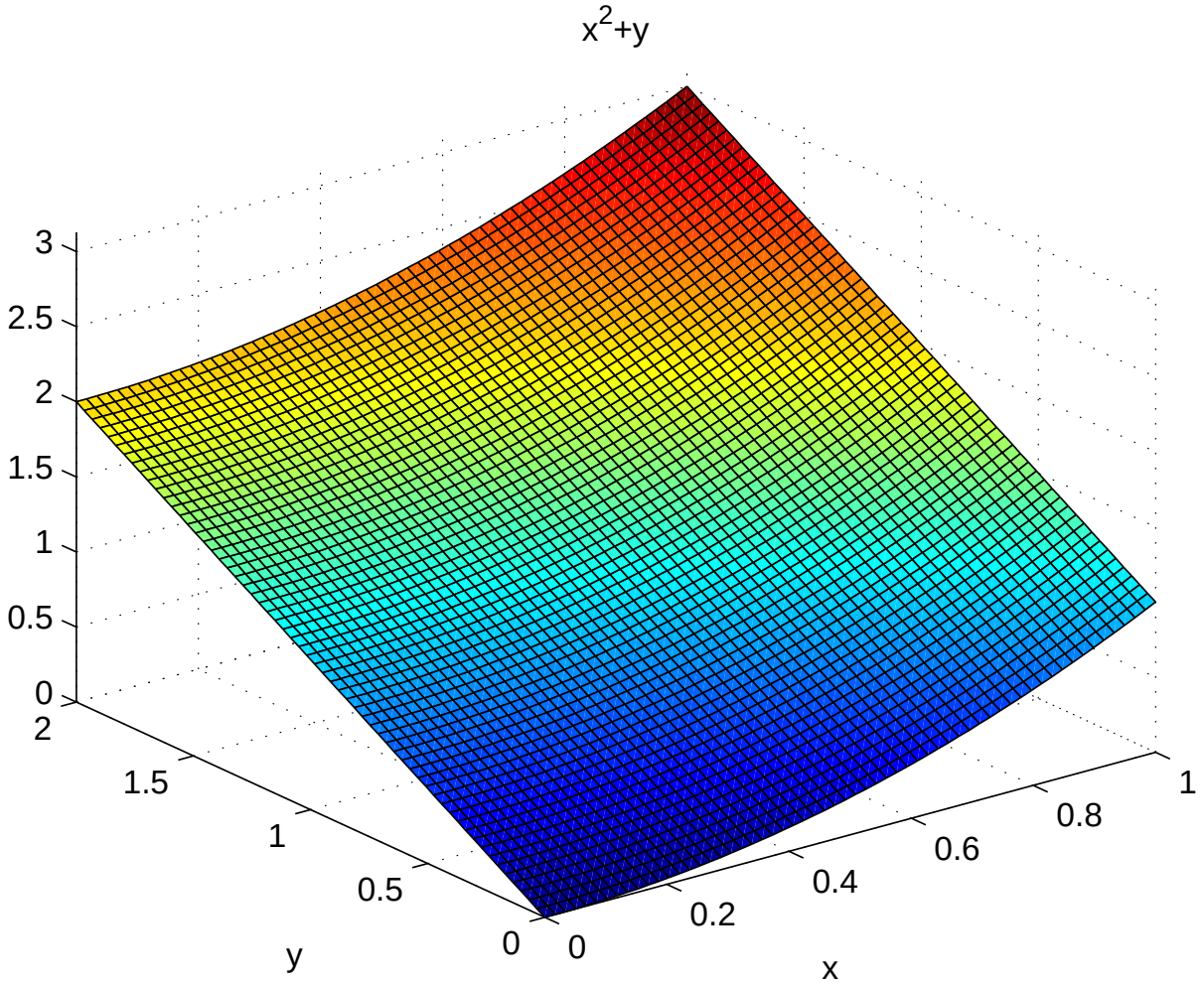
3D-grafer (forts.)

- *Exempel.* Plotta ytan $z = f(x, y) = x^2 + y$ i området $0 \leq x \leq 1$, $0 \leq y \leq 2$.

  Vi visar först hur ytan kan ritas med funktionen `ezsurf`:

  (`ez` = easy to use, `surf` = surface)

```
>> f = @(x,y) x.^2 + y    % Anonym funktion. Elementvisa beräkningar
f =
    @(x,y)x.^2+y

>> f(1,2) % Testa att beräkna ett funktionsvärde
ans =
     3

>> ezsurf(f, [0,1,0,2])   % Argument 2: [xmin,xmax,ymin,ymax]
```


x^2+y
3
2.5
2
1.5
1
0.5
0
2
1.5
1
0.5
0
y
0
0.2
0.4
0.6
0.8
1
x

3D-grafer (forts.)

- För större flexibilitet kan man använda funktioner på lägre nivå:
  1. Generera en *grid* (rutnät) av punkter (x_{ij}, y_{ij}) med kommandot `meshgrid`.
  2. Beräkna $z_{ij} = f(x_{ij}, y_{ij})$ för varje gridpunkt (x_{ij}, y_{ij}).
  3. Förbind punkterna (x_{ij}, y_{ij}, z_{ij}) med kommandot `surf` (eller `mesh`).

3D-grafer (forts.)

- *Exempel.* (forts.)
- *Steg 1.* Vi skapar två matriser X och Y som innehåller samtliga x- respektive y-koordinater.

```
>> x = 0:0.25:1;   % Nodpunkter i x-led.
>> y = 0:0.25:2;   % Nodpunkter i y-led.
>> [X,Y] = meshgrid(x,y);
```

Nodpunkterna definierar följande grid av (x_{ij}, y_{ij})-par (med y-axeln riktad "nedåt"):

```
(0,0)       (0.25,0)     (0.5,0)      (0.75,0)     (1,0)
(0,0.25)    (0.25,0.25)  (0.5,0.25)   (0.75,0.25)  (1,0.25)
(0,0.5)     (0.25,0.5)   (0.5,0.5)    (0.75,0.5)   (1,0.5)
 ...          ...          ...          ...          ...
(0,2)       (0.25,2)     (0.5,2)      (0.75,2)     (1,2)
```

3D-grafer (forts.)

- *Exempel.* (forts.)
- *Steg 1.* (forts.) Matrisen X innehåller x-koordinaterna x_{ij} för samtliga gridpunkter:

```
>> X
X =
         0    0.2500    0.5000    0.7500    1.0000
         0    0.2500    0.5000    0.7500    1.0000
         0    0.2500    0.5000    0.7500    1.0000
         0    0.2500    0.5000    0.7500    1.0000
         0    0.2500    0.5000    0.7500    1.0000
         0    0.2500    0.5000    0.7500    1.0000
         0    0.2500    0.5000    0.7500    1.0000
         0    0.2500    0.5000    0.7500    1.0000
         0    0.2500    0.5000    0.7500    1.0000
         0    0.2500    0.5000    0.7500    1.0000
```

3D-grafer (forts.)

- *Exempel.* (forts.)
- *Steg 1.* (forts.) Matrisen Y innehåller y-koordinaterna y_{ij} för samtliga gridpunkter:

```
>> Y
Y =
         0         0         0         0         0
    0.2500    0.2500    0.2500    0.2500    0.2500
    0.5000    0.5000    0.5000    0.5000    0.5000
    0.7500    0.7500    0.7500    0.7500    0.7500
    1.0000    1.0000    1.0000    1.0000    1.0000
    1.2500    1.2500    1.2500    1.2500    1.2500
    1.5000    1.5000    1.5000    1.5000    1.5000
    1.7500    1.7500    1.7500    1.7500    1.7500
    2.0000    2.0000    2.0000    2.0000    2.0000
```

3D-grafer (forts.)

- *Exempel.* (forts.)
- *Steg 2.* Nu beräknar vi en matris Z som innehåller funktionsvärdena $z_{ij} = f(x_{ij}, y_{ij}) = x_{ij}^2 + y_{ij}$ för samtliga gridpunkter:

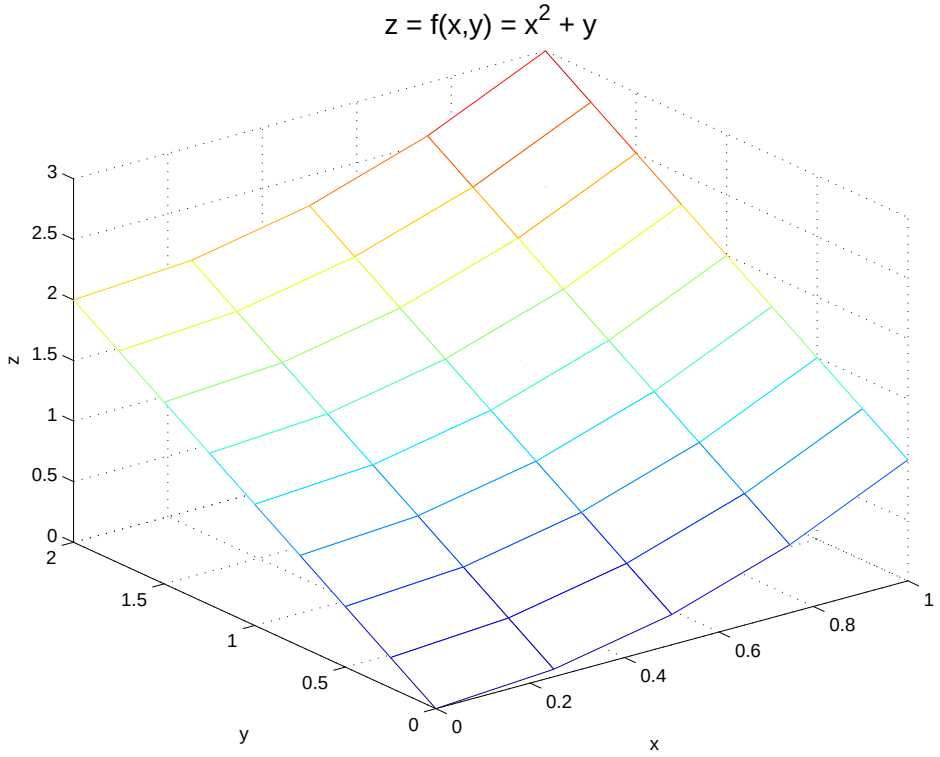
```
>> Z = X.^2 + Y   % OBS!!! Elementvisa operationer!!!

Z =
         0    0.0625    0.2500    0.5625    1.0000
    0.2500    0.3125    0.5000    0.8125    1.2500
    0.5000    0.5625    0.7500    1.0625    1.5000
    0.7500    0.8125    1.0000    1.3125    1.7500
    1.0000    1.0625    1.2500    1.5625    2.0000
    1.2500    1.3125    1.5000    1.8125    2.2500
    1.5000    1.5625    1.7500    2.0625    2.5000
    1.7500    1.8125    2.0000    2.3125    2.7500
    2.0000    2.0625    2.2500    2.5625    3.0000
```

3D-grafer (forts.)

- *Exempel.* (forts.)
- *Steg 3.* Slutligen förbinds punkterna (x_{ij}, y_{ij}, z_{ij}) i matriserna X, Y och Z med kommandot `mesh`:

```
>> mesh(X,Y,Z)

>> xlabel('x'), ylabel('y'), zlabel('z')

>> title('z = f(x,y) = x^2 + y','FontSize',15)
```

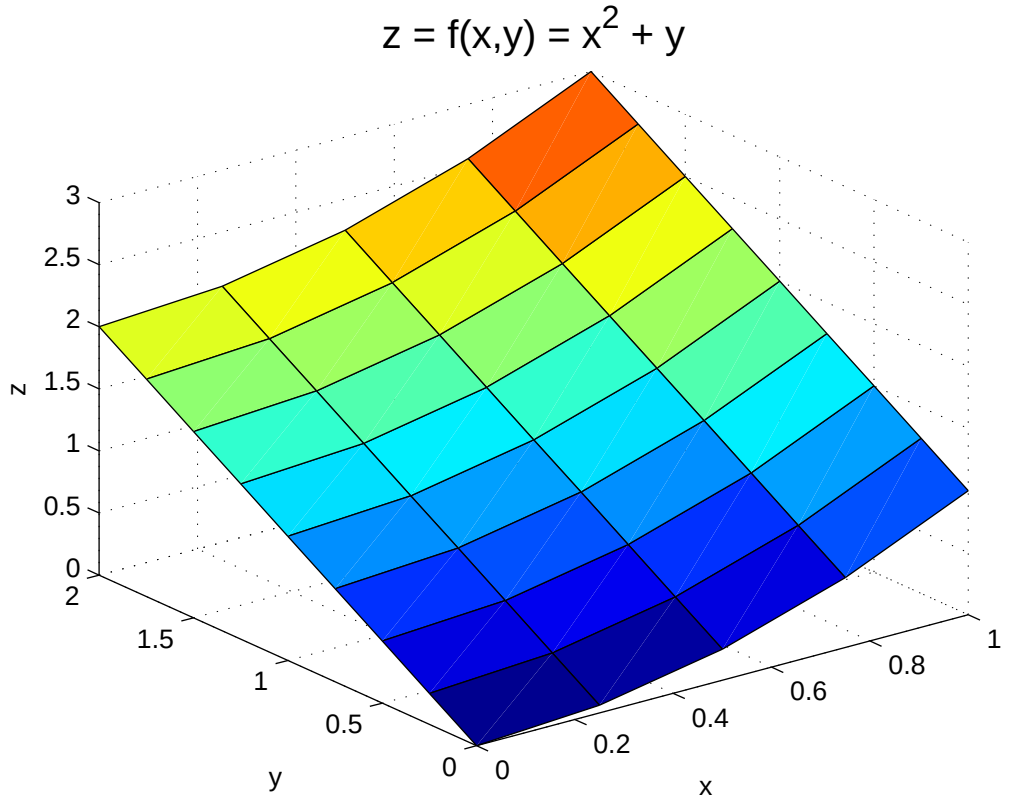
3D-grafer (forts.)



3D-grafer (forts.)

- *Exempel.* (forts.)
- *Steg 3.* Vill man färglägga ytan kan man använda kommandot `surf`:

```
>> surf(X,Y,Z)

>> xlabel('x'), ylabel('y'), zlabel('z')

>> title('z = f(x,y) = x^2 + y','FontSize',15)
```


z = f(x,y) = x^2 + y
3
2.5
2
1.5
1
0.5
0
z
2
1.5
1
0.5
0
y
0
0.2
0.4
0.6
0.8
1
x

3D-grafer (forts.)

- *Syntax.*

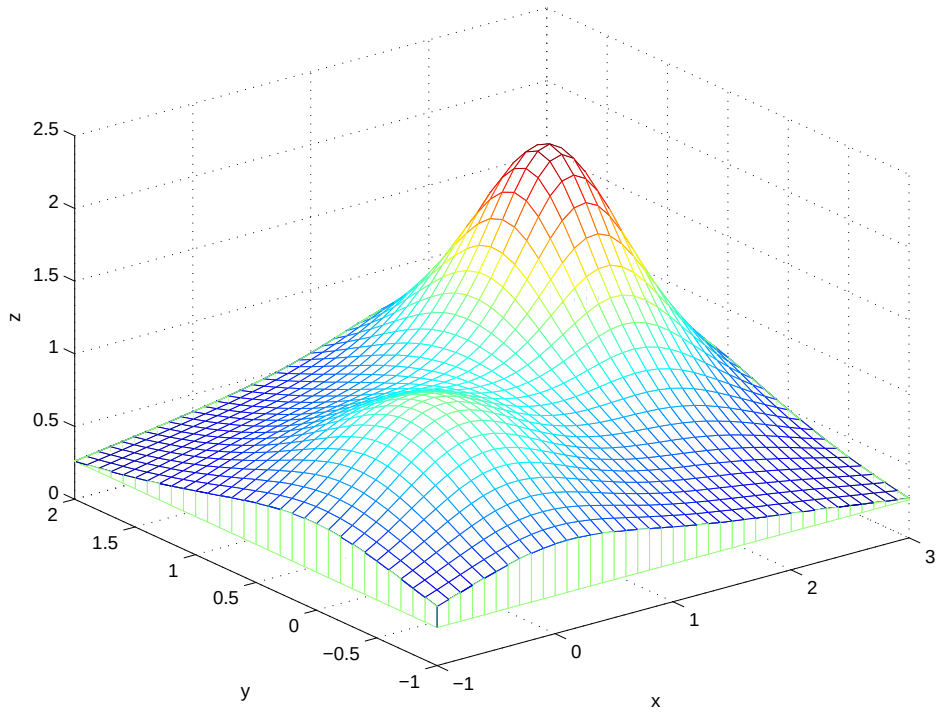
`[X,Y] = meshgrid(x,y)`	Genererar gridmatriser X och Y för området $a \leq x \leq b$, $c \leq y \leq d$ givet vektorer x och y med nodpunkter i intervallen $[a, b]$ respektive $[c, d]$.
`mesh(X,Y,Z)`	Plottar ytan given av punkterna (x_{ij}, y_{ij}, z_{ij}) i matriserna X, Y och Z. Sammanbinder punkterna.
`surf(X,Y,Z)`	Plottar ytan given av punkterna (x_{ij}, y_{ij}, z_{ij}) i matriserna X, Y och Z. Färglägger också ytan.
`meshz(X,Y,Z)`	Fungerar som `mesh` men ritar dessutom referenslinjer till xy-planet.
`meshc(X,Y,Z)`	Fungerar som `mesh` men ritar dessutom nivåkurvor i xy-planet.

3D-grafer (forts.)

- *Exempel.* Plotta ytan $z = f(x,y) = \frac{1}{x^2+y^2+1} + \frac{1}{(x-2)^2+(y-1)^2+0.5}$ i området $-1 \leq x \leq 3$, $-1 \leq y \leq 2$.

```
>> x = -1:0.1:3; y = -1:0.1:2;

>> [X,Y] = meshgrid(x,y);

>> Z = 1./(X.^2 + Y.^2 + 1) + 1./((X-2).^2 + (Y-1).^2 + 0.5);

>> meshz(X,Y,Z)

>> xlabel('x'), ylabel('y'), zlabel('z')
```

3D-grafer (forts.)



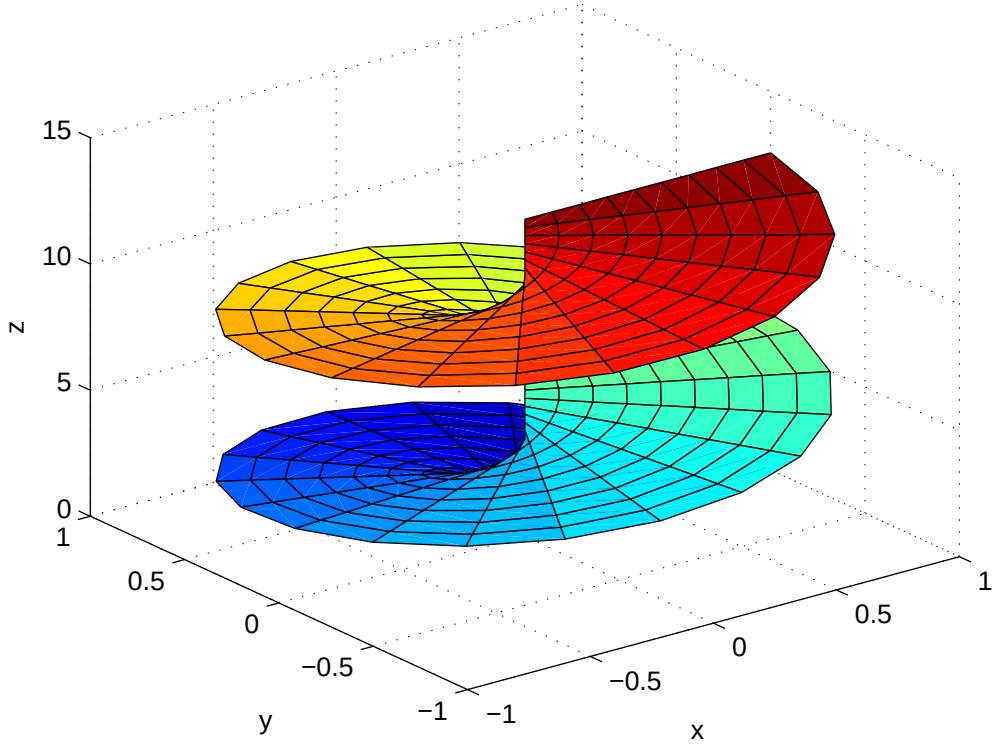
3D-grafer (forts.)

- *Exempel.* Plotta ytan (spiralytan):

$$\begin{cases} x = s\cos(t) \\ y = s\sin(t) \\ z = t \end{cases}, \quad 0 \le s \le 1,\ 0 \le t \le 4\pi.$$

```
>> s = linspace(0,1,10);       % 10 punkter
>> t = linspace(0,4*pi,40);    % 40 punkter
>> [S,T] = meshgrid(s,t);
>> X = S.*cos(T);
>> Y = S.*sin(T);
>> Z = T;
>> surf(X,Y,Z)
>> xlabel('x'), ylabel('y'), zlabel('z')
```

3D-grafer (forts.)



Nivåkurvor

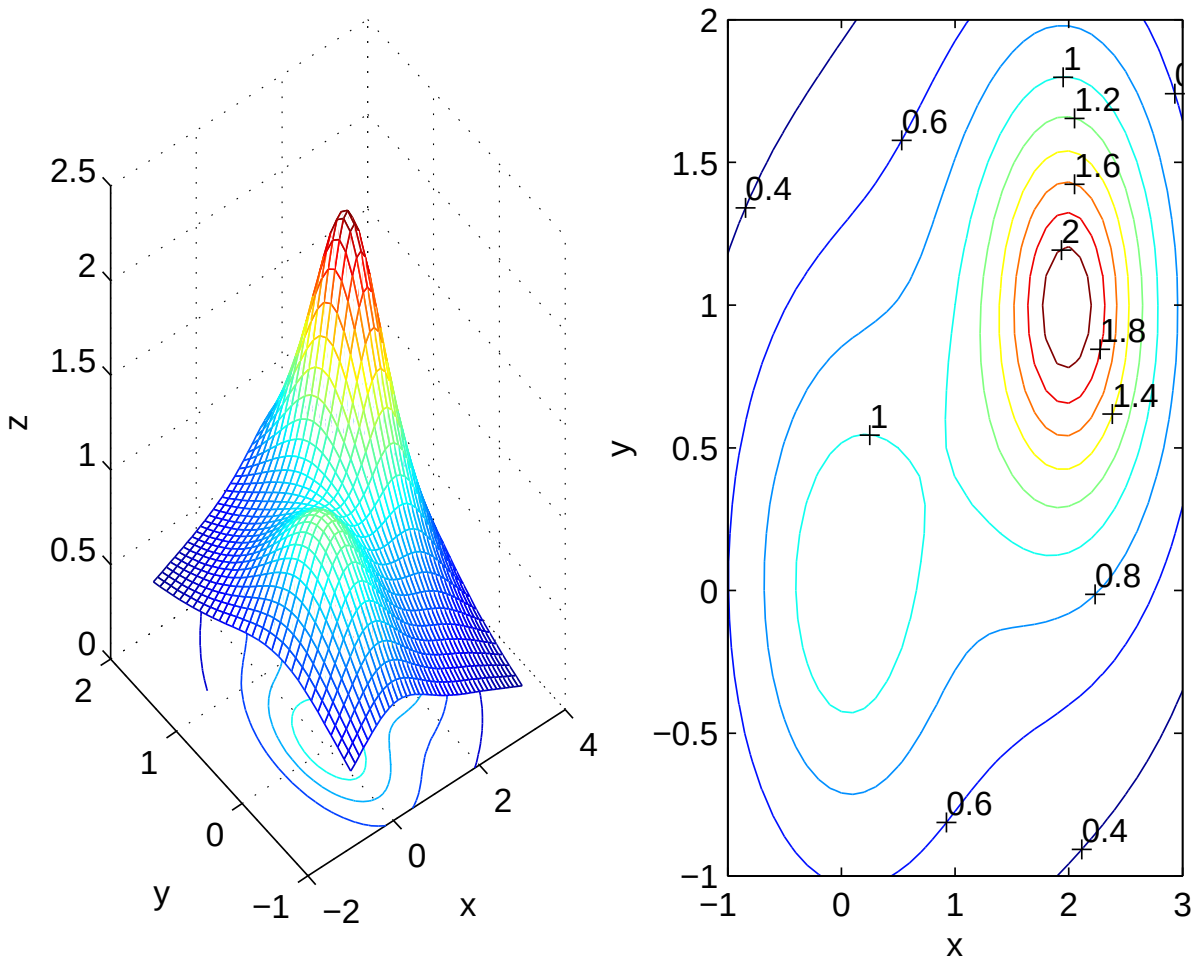
- Nivåkurvor till en yta, d.v.s. kurvor i xy-planet längs vilka z är *konstant*, kan ritas med hjälp av kommandot `contour`.
- *Syntax.*

`contour(X,Y,Z)`	Ger nivåkurvor till ytan definierad av matriserna X, Y och Z.
`contour(X,Y,Z,n)`	Ger n nivåkurvor.
`contour(X,Y,Z,v)`	Ger nivåkurvor för nivåerna givna i vektorn $v = (v_1, v_2, ..., v_n)$.
`contour(X,Y,Z,[v1 v1])`	Ger en enda nivåkurva motsvarande $z = v1$.
`C = contour(...)`	Ger nivåkurvor samt en konturmatris med vilken man kan skriva ut nivåangivelser.
`clabel(C)`	Skriver ut nivåangivelser.

Nivåkurvor (forts.)

- *Exempel.* Plotta ytan $z = f(x,y) = \frac{1}{x^2+y^2+1} + \frac{1}{(x-2)^2+(y-1)^2+0.5}$ i området $-1 \leq x \leq 3$, $-1 \leq y \leq 2$. Rita nivåkurvor.

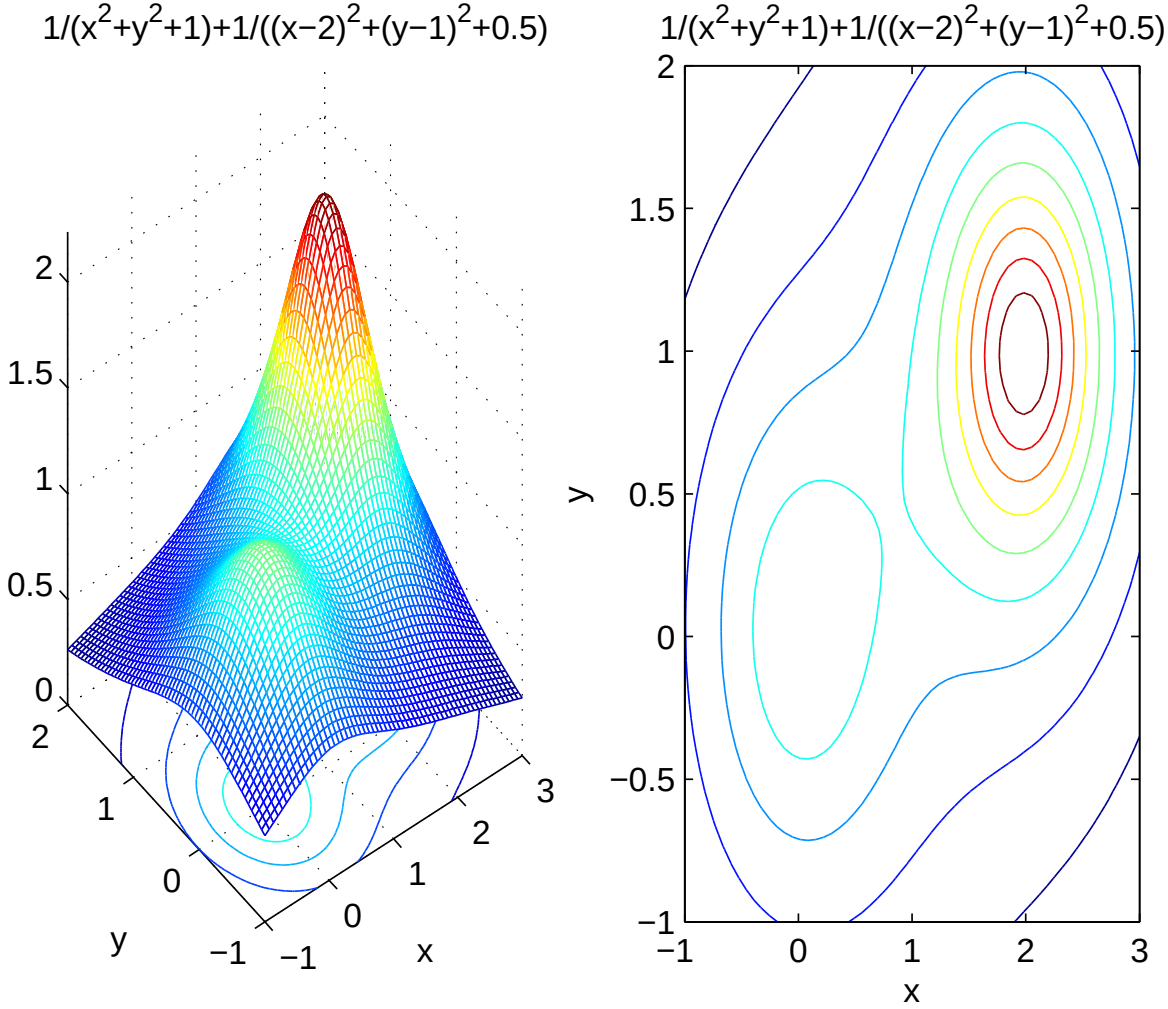
```
>> x = -1:0.1:3; y = -1:0.1:2;
>> [X,Y] = meshgrid(x,y);
>> Z = 1./(X.^2 + Y.^2 + 1) + 1./((X-2).^2 + (Y-1).^2 + 0.5);

>> subplot(1,2,1)
>> meshc(X,Y,Z)
>> xlabel('x'), ylabel('y'), zlabel('z')

>> subplot(1,2,2)
>> C = contour(X,Y,Z);
>> clabel(C)
>> xlabel('x'), ylabel('y')
```


2.5
2
1.5
1
0.5
0
z
2
1
0
y
−1
−2
0
2
4
x
2
1.5
1
0.5
0
−0.5
−1
y
−1
0
1
2
3
x
0.4
0.6
1
1.2
1.6
2
1.8
1.4
1
0.8
0.6
0.4

Nivåkurvor (forts.)

- Det finns `ez` (easy to use)-varianter också av `contour` och `meshc`.
- Vi gör om förra exemplet:
- *Exempel.* Plotta ytan $z = f(x,y) = \frac{1}{x^2+y^2+1} + \frac{1}{(x-2)^2+(y-1)^2+0.5}$ i området $-1 \leq x \leq 3$, $-1 \leq y \leq 2$. Rita nivåkurvor.

```
>> f = @(x,y) 1./(x.^2 + y.^2 + 1) + 1./((x-2).^2 + (y-1).^2 + 0.5)

>> subplot(1,2,1)
>> ezmeshc(f, [-1,3,-1,2])

>> subplot(1,2,2)
>> ezcontour(f, [-1,3,-1,2])
```

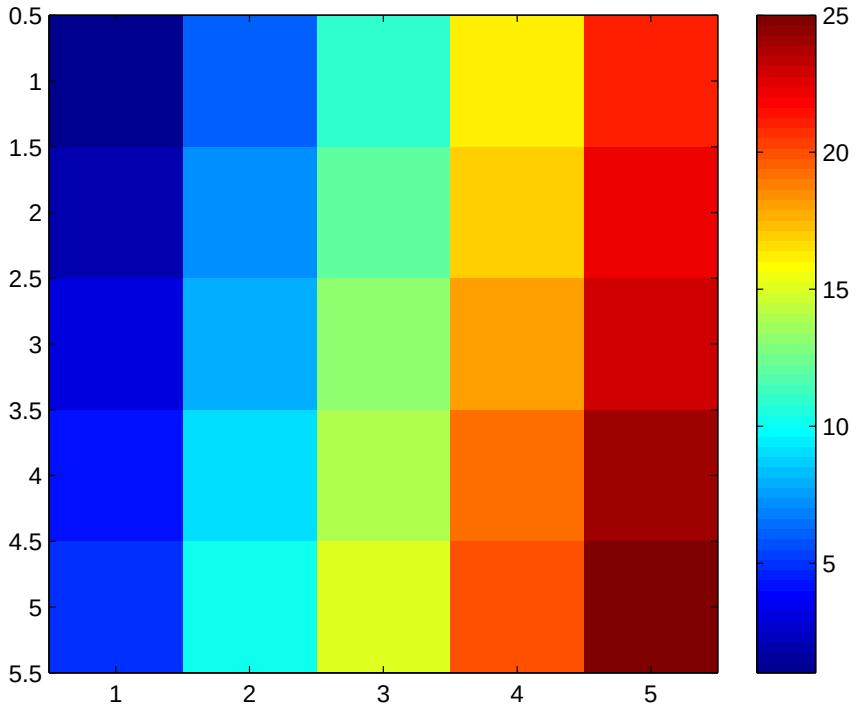

1/(x^2+y^2+1)+1/((x−2)^2+(y−1)^2+0.5)
2
1.5
1
0.5
0
2
1
0
y
−1
−1
0
1
2
3
x
1/(x^2+y^2+1)+1/((x−2)^2+(y−1)^2+0.5)
2
1.5
1
0.5
0
−0.5
−1
y
−1
0
1
2
3
x

Matriser och bilder

- Matriser kan visas som bilder med kommandot `imagesc`.
- *Exempel.*

```
>> A = reshape(1:25,5,5)  % Omforma vektorn [1 2 3 ... 25]
                          % till en 5x5-matris
A =
     1     6    11    16    21
     2     7    12    17    22
     3     8    13    18    23
     4     9    14    19    24
     5    10    15    20    25

>> imagesc(A)

>> colorbar('vert')   % Ger förklarande färgkarta i vertikal stapel.
```

Matriser och bilder (forts.)



```
>> map = colormap   % Ger färgkartans matris. Varje rad är en RGB-trippel.
map =
         0         0    0.5625     % Bara blått. (Minsta elementvärdet: 1)
         0         0    0.6250
...
         0         0    0.9375
         0         0    1.0000
         0    0.0625    1.0000     % Grönt och blått.
         0    0.1250    1.0000
...
    1.0000    0.1250         0     % Rött och grönt.
    1.0000    0.0625         0
    1.0000         0         0
    0.9375         0         0
...
    0.5625         0         0
    0.5000         0         0     % Bara rött. (Största elementvärdet: 25)

>> size(map)
ans =
    64     3   % 64 stycken färger.
```

Matriser och bilder (forts.)

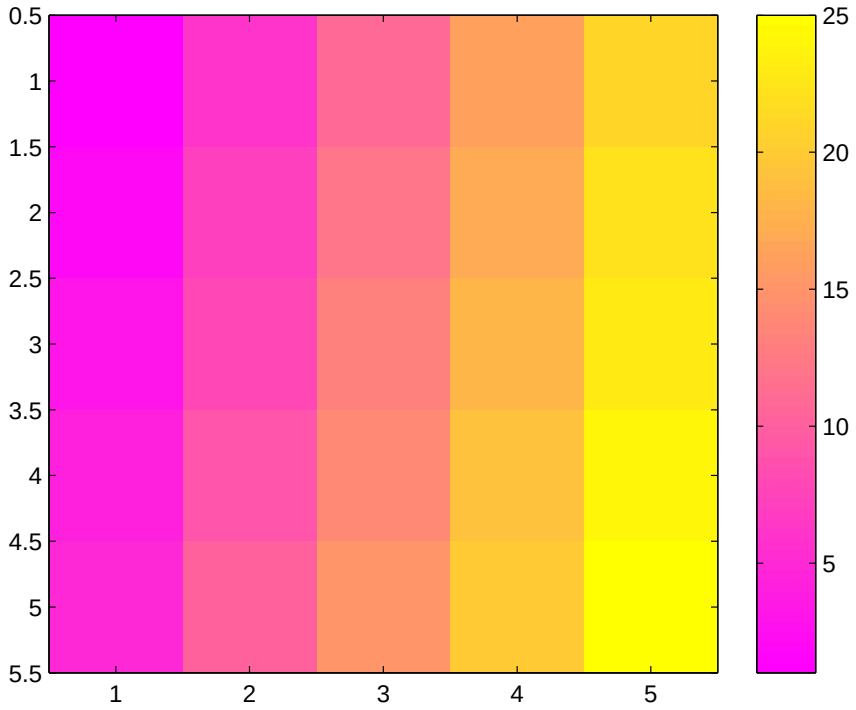
- Man kan också byta färgkarta med kommandot `colormap`.
- Exempel på färgkartor: *jet* (default), *gray*, *copper*, *hot*, *cool*, *spring*, ...
- Ge kommandot

  ```
  >> help graph3d
  ```

  för fler exempel.
- *Exempel.*

```
>> A = reshape(1:25,5,5);   % Samma matris som ovan!
>> colormap('spring')
>> imagesc(A)
>> colorbar('vert')
```

Matriser och bilder (forts.)

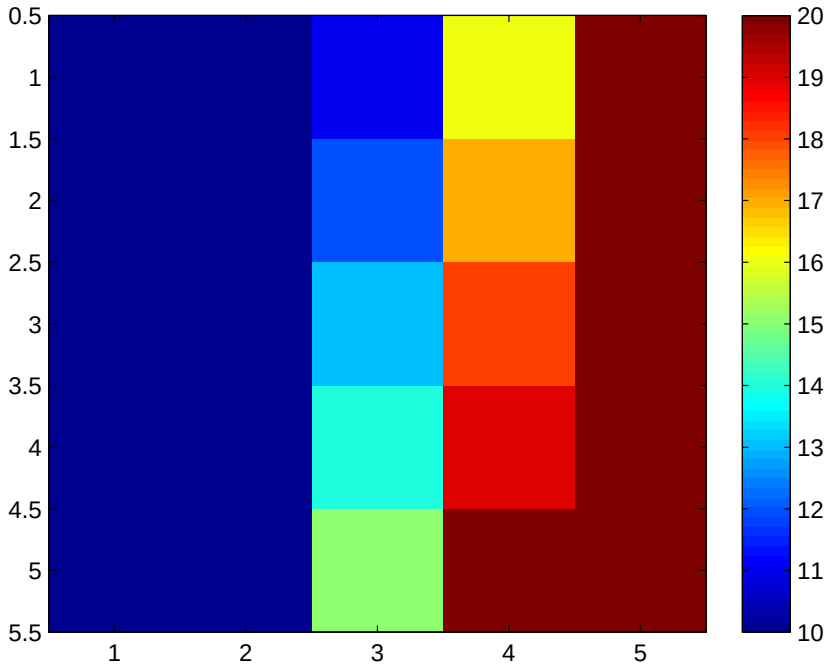


Matriser och bilder (forts.)

- Det går att endast sprida ut elementvärden i ett visst intervall `[low,high]` i färgkartan.
- Värden utanför intervallet får den första eller sista färgen i färgkartan.
- *Exempel.*

```
>> colormap('jet')       % Tillbaka till default.
>> imagesc(A, [10,20])
>> colorbar('vert')
```

Matriser och bilder (forts.)



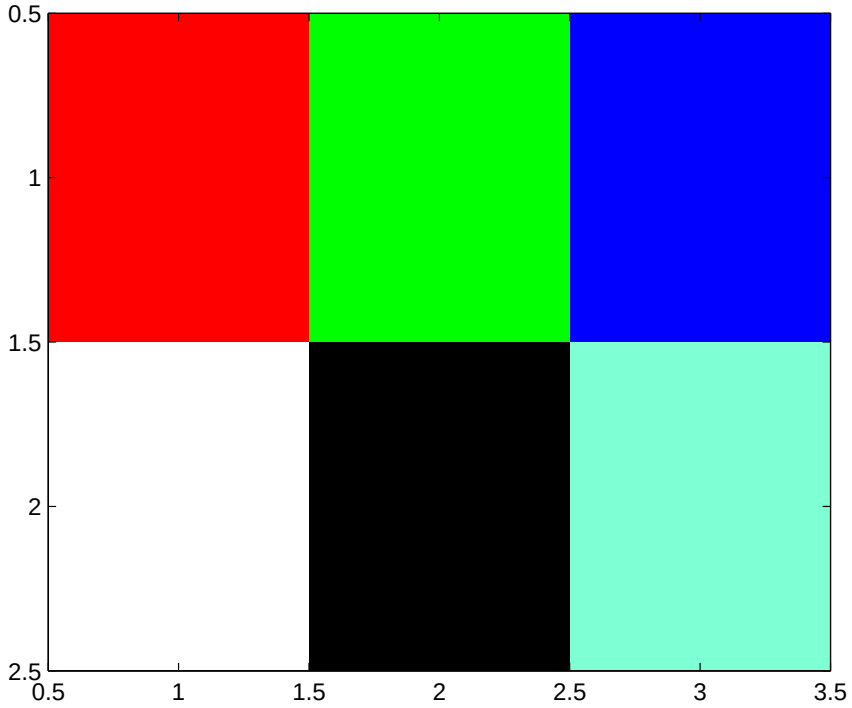
Matriser och bilder (forts.)

- *Exempel.* Tredimensionell matris.

```
>> B(1,1,:) = [1 0 0]; % Röd
>> B(1,2,:) = [0 1 0]; % Grön
>> B(1,3,:) = [0 0 1]; % Blå
>> B(2,1,:) = [1 1 1]; % Vit
>> B(2,2,:) = [0 0 0]; % Svart
>> B(2,3,:) = [127/255 1 212/255]; % Akvamarin
>> imagesc(B)
```

  En färg specificeras med tre tal mellan 0 och 1, en s.k. *"rgb-trippel"*. Det första talet anger intensiteten av *rött ljus*, det andra av *grönt ljus* och det tredje av *blått ljus*.

Matriser och bilder (forts.)



Matriser och bilder (forts.)

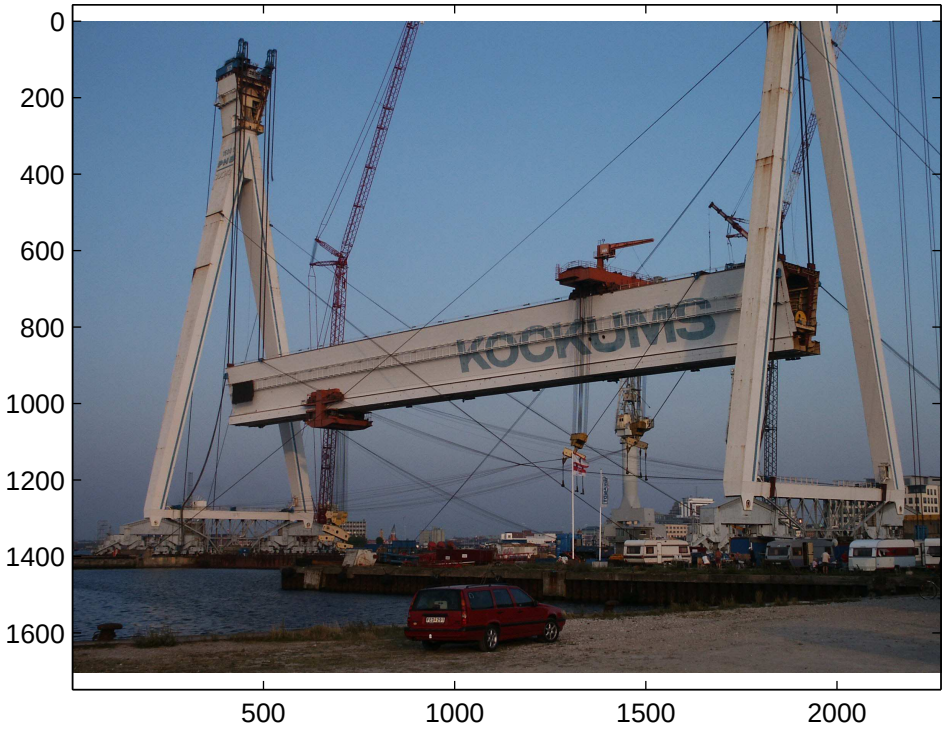
- Bilder kan läsas in till MATLAB med kommandot `imread`.

  Exempel. Läs in en jpg-bild.

```
>> A = imread('pict0012','jpg');

>> whos A     % Antal pixlar: 1704 x 2272
  Name      Size                         Bytes  Class
  A      1704x2272x3                  11614464  uint8 array

>> imagesc(A), axis equal
```

- OBS! Här lagras varje färg som tre *heltal* (8 bitars = 1 bytes heltal utan tecken: `uint8`) mellan 0 och 255.

Matriser och bilder (forts.)



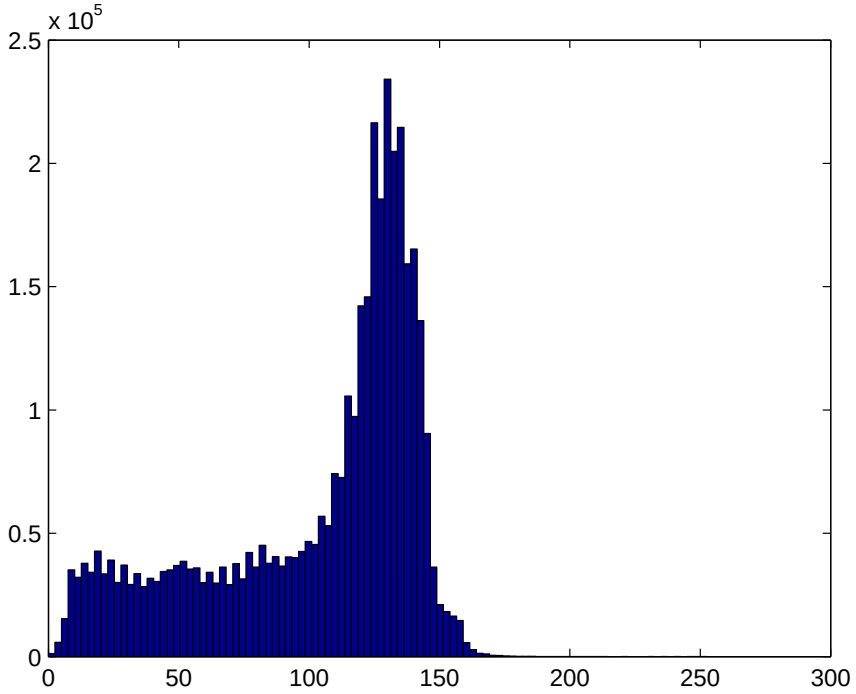
- MATLAB kan utföra vissa aritmetiska heltalsoperationer men om man vill göra någon form av bildbehandling är det bäst att först göra om matriselementen till *double*.

```
>> B = double(A);

>> whos B   % Går åt 8 ggr så mycket lagringsutrymme!
  Name      Size                          Bytes  Class
  B      1704x2272x3                    92915712  double array

>> I = 0.2989*B(:,:,1) + ...
       0.5870*B(:,:,2) + ...  % Skapa en 1704x2272-matris I genom
       0.1140*B(:,:,3);    % att vikta samman RGB-ljusintensiteterna.
                           % Vikter anpassade till ögats känslighet
                           % enligt NTSC-standard.
>> colormap('gray'), imagesc(I), axis equal  % Ger gråskalebild.
   % Färgkartan 'gray' innehåller gråskalor från svart till vitt.
```

Matriser och bilder (forts.)

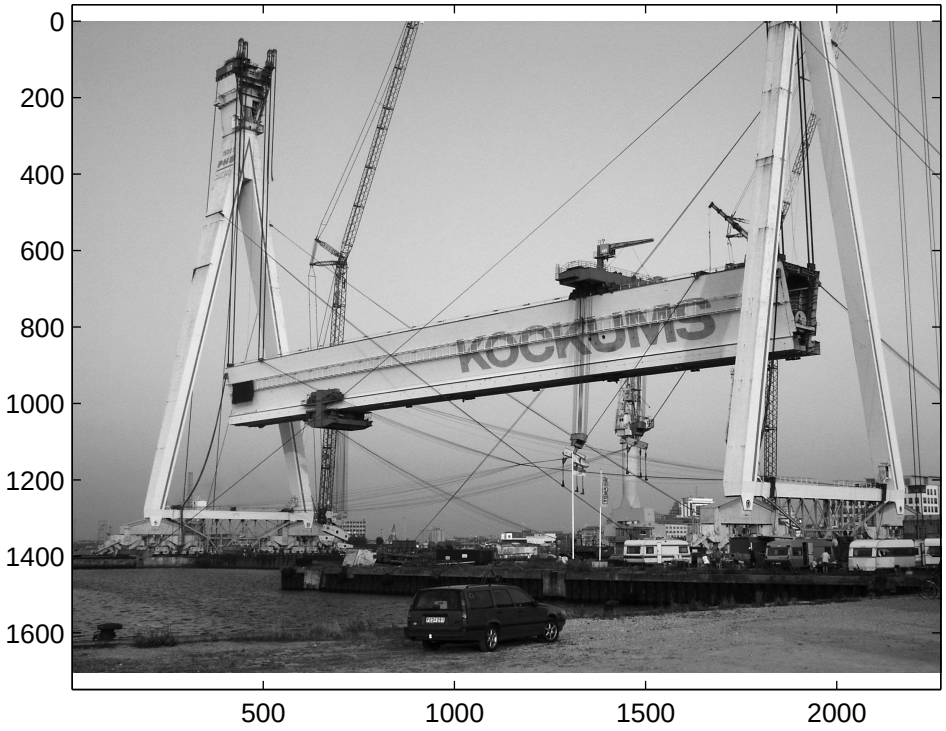
- Vi ritar ett histogram över de sammanviktade ljusintensiterna:

```
>> r = reshape(I, [1 1704*2272]);   % Omforma I till en radvektor.
>> hist(r,100)  % 100 klasser.
```



- Vi ser att det är mycket få pixlar som har en intensitet över 175 och utnyttjar detta för att bättre utnyttja färgkartan (ger bättre kontrast):

```
>> colormap('gray'), imagesc(I, [0,175]), axis equal
```



Övningar

1. Simulera 1000 tärningskast i MATLAB. Rita ett histogram.

Tips. Ett tärningskast kan simuleras med

```
>> ceil(6*rand(1,1))
```

2. Plotta kurvan: $\begin{cases} x &= t\cos(t) \\ y &= t\sin(t) \\ z &= at \end{cases}$, $0 \leq t \leq 6\pi$.

Prova med några olika värden på konstanten a.

3. Plotta ytan $z = f(x,y) = x - y^2$, $-1 \leq x \leq 1$, $-1 \leq y \leq 1$.

Rita nivåkurvor.

4. Plotta ytan $z = f(x,y) = x^2 + y^2$, $-1 \leq x \leq 1$, $-1 \leq y \leq 1$.

Rita nivåkurvor.

5. Plotta ytan (enhetssfären):

$$\begin{cases} x &= \sin(s)\cos(t) \\ y &= \sin(s)\sin(t) \\ z &= \cos(s) \end{cases}, \quad 0 \le s \le \pi, \quad 0 \le t \le 2\pi.$$

Rita nivåkurvor.

Om du läst om bilder:

6. Ge följande kommandon:

```
>> load clown  % Läser in två matriser:
               %     X (bild)
               %   map (färgkarta)
>> imagesc(X), colormap(map)
```

Prova därefter att byta färgkarta.

7. Läs in en egen bild (t.ex. jpg) med `imread`. Visa bilden i MATLAB med kommandot `imagesc`. Titta i Workspace Browser hur många pixlar bilden har.